

WO 02/06949 A2

10 Verfahren und Vorrichtung zur Messung der Laufzeit einer
Task in einem Echtzeitsystem

Stand der Technik

15 Die vorliegende Erfindung betrifft ein Verfahren und eine
Vorrichtung zur Messung der Laufzeit einer Task in einem
Echtzeitsystem, insbesondere einem Echtzeitbetriebssystem.
Als typische Echtzeitbetriebssysteme seien beispielsweise
RCOS oder ERCOS genannt.

20 Unter einer Task versteht man einen ablauffähigen
Programmteil innerhalb eines Echtzeitsystems. Die Summe
aller Tasks ergibt das gesamte Echtzeitsystem. Man spricht
dann auch von einem Multitaskingsystem. Ein Echtzeitssystem
25 ist ein Computerprogramm, an dessen Zeitverhalten strenge
Anforderungen gerichtet sind. So muß sichergestellt sein,
daß das Programm innerhalb einer vorgegebenen Zeitspanne
auf bestimmte Ereignisse reagiert. Echtzeitsysteme und
insbesondere Echtzeitbetriebssysteme finden vorwiegend bei
30 sicherheitskritischen Systemen Anwendung.

Um ein gewünschtes zeitliches Verhalten des gesamten
Systems sicherzustellen, muß gewährleistet sein, daß jede
Task bestimmte zeitliche Vorgaben einhält. Die Tasks in

einem Echtzeitsystem besitzen dabei unterschiedliche Zeitanforderungen, sogenannte Deadlines. Diese stellen sich als unterschiedliche Prioritäten dar. Es wird angenommen, daß Tasks höherer Priorität solche mit niedrigerer

5 Priorität unterbrechen können. Das bedeutet, daß sobald eine höherwertige Task zum Ablauf kommt, diese den Ablauf einer niederwertigeren Task unterbricht.

Um das Zeitverhalten des gesamten Systems zu bestimmen und

10 gegebenenfalls zu modifizieren, ist es notwendig, die Laufzeit jeder einzelnen Task im Laufe der Programmentwicklung, in der Testphase und auch während des Betriebs des gesamten Systems, insbesondere bei nachträglichen Programmänderungen, zu messen.

15 Bekannt ist es, die Laufzeit einer Task manuell zu ermitteln. Dies wird beispielsweise mit Hilfe eines CPU-Emulators durchgeführt. Hierbei stellt sich das Problem, daß es beim Messen der Laufzeit einer Task vorkommt, daß

20 diese durch eine Task mit höherer Priorität unterbrochen wird (preemptives Multitasking). Die gemessene Zeit wird verfälscht, da nicht mehr nur die Laufzeit einer bestimmten Task gemessen wird. Außerdem ist die manuelle Messung sehr zeitaufwendig. Dies erhöht die Kosten und die Dauer der

25 Entwicklung des Programms.

Eine weiteres bekanntes Verfahren zur Messung der Laufzeit einer Task sieht vor, in jeder Task einen Zeitmesser mitlaufen zu lassen. Als Zeitmesser kann beispielsweise ein

30 hochauflösender Hardware-Timer verwendet werden. Aber auch eine reine Softwarelösung ist denkbar. Die Schrittweite des Zeitmessers muß dabei viel kleiner sein als die üblichen Tasklaufzeiten. Auf diese Weise ist zwar sichergestellt, daß die Zeitmessung immer bei Taskbeginn gestartet wird,

doch kann mit dieser Methode nur die Bruttolaufzeit der Task, das heißt die eigentliche Laufzeit oder Nettolaufzeit der Task plus die Unterbrechungen durch höherwertigere Tasks gemessen werden. Diese Methode ist zwar weniger
5 arbeitsaufwendig als das zuvor beschriebene manuelle Verfahren, doch sind die ermittelten Ergebnisse nicht wesentlich exakter. Daher ergibt sich auch kein entscheidender Vorteil gegenüber der manuellen Methode.

- 10 Aus der EP 0 534 884 A1 ist ein Verfahren zum Überprüfen des Zeitverhaltens eines Multitaskingsystems bekannt. Das Verfahren sieht vor, daß die Laufzeit einzelner Tasks gemessen und gegebenenfalls bei Überschreiten einer vorgebbaren Zeitdauer ein Alarm ausgelöst wird. Die
15 Zeitdauer wird mit Hilfe eines Timers bzw. eines Zählers bestimmt. Dabei wird die Zeitdauer als Vielfaches des Zyklusses des Prozessors bestimmt. Das Multitaskingsystem weist eine Anzahl von Tasks auf, denen allen eine bestimmte Priorität zugeordnet ist. So kann es vorkommen, daß eine
20 niederwertige Task von einer höherwertigeren Task unterbrochen wird. Um sicherzustellen, daß das gesamte Programm die notwendigen Zeitanforderungen einhält, muß es möglich sein, die Laufzeit jeder einzelnen Task zu bestimmen.

- 25 Zu Beginn des Ablaufs einer Task, deren Laufzeit in Prozessorzyklen bestimmt werden soll, wird der Zähler gestartet. Der Zähler beginnt bei einem vorgegebenen Wert und verringert nach jedem Zyklus seinen Zustand. Wird eine
30 bestimmte untere Grenze erreicht, wird ein Alarmsignal ausgegeben. Wird die Task durch eine Task mit höherer Priorität unterbrochen, wird der Zähler gestoppt und der aktuelle Zustand gespeichert. Nach Beendigung der

Unterbrechung wird der Zähler beginnend vom gespeicherten Zustand erneut gestartet.

Nachteilig bei dem in der EP 0 534 884 A1 beschriebenen
5 Verfahren ist es, daß immer nur die Laufzeit einer Task
während eines Meßdurchgangs gemessen werden kann. Erst wenn
die Laufzeit der Task bestimmt ist, d. h. die Task beendet
ist, kann mit dem Zähler die Laufzeit einer weiteren Task
bestimmt werden. Ein Meßdurchgang beginnt mit Beginn des
10 Ablaufs einer Task und deren Beendigung.

Aufgabe der vorliegenden Erfindung ist es daher, ein
Verfahren und eine Vorrichtung vorzuschlagen, mit denen
eine exakte Messung der Laufzeit einer Task möglich ist und
15 außerdem in einem Meßdurchgang die Laufzeiten mehrerer
Tasks bestimmt werden können.

Die Aufgabe bezüglich des Verfahrens wird durch ein
Verfahren mit den Merkmalen des Anspruchs 1 gelöst.
20

Die auf die Vorrichtung bezogene Aufgabe wird durch eine
solche mit den Merkmalen des Anspruchs 6 gelöst.

Vorteilhafte Ausgestaltungen der Erfindung ergeben sich aus
25 den Unteransprüchen.

Vorteile der Erfindung

Das erfindungsgemäße Verfahren dient zur Messung der
30 Laufzeit einer Task in einem Echtzeitsystem mit einer
Anzahl von Tasks, welche wenigstens teilweise
unterschiedlichen Prioritätsebenen zugeordnet sind. Dabei
können die einzelnen Tasks alle in unterschiedlichen
Prioritätsebenen ablaufen. Es ist aber auch möglich das

einige oder auch alle Tasks in derselben Prioritätsebene bzw. denselben Prioritätsebenen ablaufen. Für die Zeitmessung der Laufzeit ist ein Zeitmesser vorgesehen, dessen Zustand die Laufzeit wiedergibt. Bei Beginn einer Task, deren Laufzeit bestimmt werden soll, wird der 5 Zeitmesser gestartet. Kommt es zu einer Unterbrechung der Task durch eine Task mit höherer Priorität, d. h. zu einem Wechsel der Prioritätsebene, wird der Zeitmesser gestoppt und der aktuelle Zustand gespeichert. Nach Beendigung der 10 Unterbrechung, sobald also die Task, deren Laufzeit bestimmt werden soll, erneut läuft, wird der gespeicherte Zustand des Zeitmessers restauriert und der Zeitmesser beginnend beim gespeicherten Zustand wieder gestartet.

15 Zweckmäßigerweise wird der Zustand des Zeitmessers bei einer Unterbrechung einer ersten Task möglichst früh gespeichert, und nach Beendigung der Unterbrechung möglichst spät, d. h. möglichst unmittelbar vor dem Wiederstart der Task, restauriert. Mit dieser Maßgabe sind 20 besonders genaue Messungen der Nettolaufzeiten einer Task möglich. Dies erweist sich als vorteilhaft, da nicht auszuschließen ist, daß es zu einem systematischen Meßfehler kommen kann, da ein Teil der Prioritätsumschaltung mitgemessen wird, bis der Zustand des 25 Zeitmessers gesichert ist. Gleiches gilt für das Restaurieren eines gespeicherten Zustandes des Zeitmessers. Dadurch, daß der Zustand des Zeitmessers bei einer Unterbrechung so schnell wie möglich gespeichert und nach Beendigung der Unterbrechung so spät wie möglich 30 restauriert wird, läßt sich dieser Fehler minimieren. Dieser Meßfehler ist in jedem Fall deutlich kleiner als der Meßfehler bei Messung der Bruttolaufzeit.

Gemäß einer bevorzugten Ausführungsform des
erfindungsgemäßen Verfahrens wird bei Unterbrechung der
ersten Task aufgrund einer prioritätshöheren zweiten Task
die Laufzeit der zweiten Task während der Unterbrechung der
5 Laufzeit der ersten Task gemessen. Selbstverständlich ist
es hierbei möglich, während entsprechenden Unterbrechungen
der zweiten Task die Laufzeit einer dritten Task zu messen
usw. Es ist so erfindungsgemäß möglich, im wesentlichen
während eines Durchlaufs eines Echtzeitsystems die
10 auftretenden Tasks im wesentlichen gleichzeitig zu messen.

Hierdurch wird, wie bereits erwähnt, da die Zeitmessung
stets zu Beginn einer Task gestartet und während einer
Unterbrechung gestoppt wird, die Nettolaufzeit der Task
15 bzw. Tasks bestimmt.

Für sämtliche Prioritätsebenen ist zweckmäßigerweise nur
ein Zeitmesser vorgesehen. Innerhalb einer Prioritätsebene
20 laufen alle Tasks kooperativ, das bedeutet, daß sie sich
nicht gegenseitig unterbrechen. Bei einem Wechsel der
Prioritätsebene, nämlich bei Beginn einer höherwertigeren
Task, wird der Zustand der verwendeten CPU gesichert, damit
sich die CPU nach Ende der Unterbrechung im selben Zustand
25 wie vor der Unterbrechung befindet. Außerdem wird bei jedem
Prioritätswechsel der Zustand des Zeitmessers gesichert. So
können mit einem Zeitmesser in einem Meßdurchgang die
Laufzeiten mehrerer Tasks, die in unterschiedlichen
Prioritätsebenen laufen, bestimmt werden.

30

Mit dem erfindungsgemäßen Verfahren lassen sich die
Nettolaufzeiten aller Tasks im System einfacher als bisher
messen. Der zusätzliche Aufwand besteht im Starten der
Messung bei Beginn jeder Task und dem Sichern und

- 7 -

- Restaurieren des Zeitmessers bei jedem Prioritätswechsel. Alle Unterbrechungen durch andere Tasks werden dabei automatisch ausgeklammert. Ein verbessertes und einfacheres Monitoring der Systemressource Laufzeit ist so möglich. Das
- 5 Verfahren ist unabhängig vom verwendeten Betriebssystem und der zugrundeliegenden Hardware. Der Ressourcenbedarf an Zeitmessern für die Messung ist unabhängig von der Anzahl der Prioritätsebenen, da nur genau ein Zeitmesser benötigt wird. Es muß jeder Prioritätsebene nur der Platz zum
- 10 Sichern des Zustands der Messung zur Verfügung gestellt sein. Dies ist für die Implementierung wesentlich, da die Anzahl der verfügbaren Zeitmesser üblicherweise stärker eingeschränkt ist als der verfügbare Speicherplatz.
- 15 Die erfindungsgemäße Vorrichtung zur Messung der Laufzeit einer Task in einem Echtzeitsystem mit einer Anzahl von Tasks, von denen jede in einer bestimmten Prioritätsebene abläuft, weist einen Zeitmesser auf. Weiterhin sind Mittel vorgesehen, die ein Starten und Stoppen des Zeitmessers
- 20 ermöglichen. Dem Zeitmesser steht ein Speicher mit einer Kapazität zur Verfügung, die der Anzahl der Prioritätsebenen angepaßt ist.

Die Vorrichtung zur Messung der Laufzeit ist bevorzugt ein

25 Hardwaretimer. Dieser kann bereits in der CPU implementiert sein.

Vorteilhafterweise ist die Schrittweite des Zeitmessers viel kleiner als die übliche Laufzeit der Tasks. Je kleiner

30 die Schrittweite ist, umso exakter sind die Ergebnisse der Messungen.

Die Erfindung wird anhand der beigefügten Zeichnung näher erläutert. In dieser zeigt:

Figur 1 den zeitlichen Verlauf des Zustands eines Zeitmessers bei einem herkömmlichen Verfahren;

5 Figur 2 den zeitlichen Verlauf des Zustands eines Zeitmessers bei dem erfindungsgemäßen Verfahren;

Figur 3 eine erfindungsgemäße Vorrichtung in schematischer Darstellung.

10

Figur 1 zeigt den zeitlichen Verlauf des Zustands eines Zeitmessers bei einem herkömmlichen Verfahren. Zu erkennen ist ein Graph 1 mit zwei Abszissen 2 und einer ersten Ordinate 3 und einer zweiten Ordinate 4. An den beiden
15 Abszissen 2 ist die Zeit aufgetragen. Mit Hilfe der ersten Ordinate 3 läßt sich der Zustand des Zeitmessers bestimmen. Die zweite Ordinate 4 gibt die Prioritätsebene, in denen die Tasks laufen, wieder. Eine erste Gerade 5 und eine zweite Gerade 6 geben den zeitlichen Verlauf des Zustands
20 des Zeitmessers wieder.

Zum Zeitpunkt t_1 beginnt eine Task A. Deren Prioritätsebene läßt sich anhand der zweiten Ordinate 4 ablesen. Ebenfalls zum Zeitpunkt t_1 wird der Zeitmesser gestartet. Die erste
25 Gerade 5 verdeutlicht die Änderung des Zustands des Zeitmessers mit der Zeit. Zum Zeitpunkt t_2 wird die Task A durch die höherwertigere Task B unterbrochen. Diese läuft auf einer höheren Prioritätsebene. Die Änderung des Zustands des Zeitmessers über der Zeit bleibt, wie die
30 erste Gerade 5 verdeutlicht, davon unbeeinflußt. Zum Zeitpunkt t_3 endet die Task B. Nach Beendigung der Unterbrechung fährt die Task A fort. Zum Zeitpunkt t_4 endet die Task A. Der Zeitmesser wird gestoppt. Der Zustand des Zeitmessers kann an der ersten Ordinate 3 abgelesen werden.

Allerdings gibt der ermittelte Wert nicht die Nettolaufzeit sondern die Bruttolaufzeit der Task A wieder. Nach Beendigung der Task A wird der Zustand des Zeitmessers zurückgesetzt. Eine Task C, welche in der gleichen

5 Prioritätsebene läuft wie die Task A, beginnt. Der Zeitmesser wird erneut gestartet. Die zweite Gerade 6 gibt hierbei den Zustand des Zeitmessers wieder.

Figur 2 zeigt den Graphen 1 aus Figur 1, der den zeitlichen

10 Verlauf des Zustands eines Zeitmessers bei dem erfindungsgemäßen Verfahren wiedergibt. Die zeitliche Abfolge der Tasks A, B und C entspricht der Abfolge in Figur 1. Eine dritte Gerade 11, eine vierte Gerade 12, eine fünfte Gerade 13 und eine sechste Gerade 14 geben den

15 Verlauf des Zustands des Zeitmessers wieder.

Wiederum zum Zeitpunkt t_1 startet die Task A. Der Zeitmesser wird gestartet. Der Zustand ändert sich mit der Zeit entsprechend der dritten Geraden 11. Zum Zeitpunkt t_2

20 beginnt die Task B. Die Task A wird unterbrochen. Der Zustand des Zeitmessers wird gespeichert und anschließend zurückgesetzt. Während der Laufzeit der Task B gibt die vierte Gerade 12 den Zustand des Zeitmessers wieder. Nach Beendigung der Task B zum Zeitpunkt t_3 kann mit Hilfe der

25 ersten Ordinate 3 der Zustand des Zeitmessers und somit die Laufzeit der Task B bestimmt werden. Nach Beendigung der Task B zum Zeitpunkt t_3 läuft die Task A weiter, wie anhand der fünften Geraden 13 dargestellt. Der zum Zeitpunkt t_2 gespeicherte Zustand wird restauriert, so daß der

30 Zeitmesser beginnend beim gespeicherten Zustand weiterläuft. Nach Beendigung der Task A zum Zeitpunkt t_4 kann der Zustand des Zeitmessers an der ersten Ordinate 3 abgelesen werden. Dieser Zustand ist ein Maß für die Nettolaufzeit der Task A. Nach Beendigung der Task A

beginnt die Task C. Diese läuft in derselben Prioritätsebene wie die Task A und konnte bzw. kann deren Ablauf somit nicht unterbrechen. Die sechste Gerade 14 gibt den Verlauf des Zustands des Zeitmessers wieder.

5

Figur 3 zeigt in einer schematischen Darstellung eine erfindungsgemäße Vorrichtung zu Messen der Laufzeit einer Task. Die Vorrichtung weist einen Zeitmesser 15 und einen Speicher 16 auf, die über eine Datenleitung 17 miteinander verbunden sind. Über die Datenleitung 17 kann der
10 Zeitmesser 15 seinen Zustand in den Speicher 16 ablegen. Im Zeitmesser 15 sind Mittel 18 vorgesehen, die ein Starten und Stoppen des Zeitmessers 15 ermöglichen.

5

Ansprüche

10

1. Verfahren zur Messung der Laufzeit wenigstens einer Task in einem eine Anzahl von Tasks umfassenden Echtzeitsystem mittels wenigstens eines Zeitmessers (15), wobei der Zeitmesser (15) zu Beginn einer ersten zu messenden Task
15 gestartet wird und der Zustand des Zeitmessers (15) im Falle einer Unterbrechung der ersten Task gespeichert wird und bei Fortsetzung der ersten Task der gespeicherte Zustand des Zeitmessers (15) restauriert und der Zeitmesser (15) ausgehend von diesem gespeicherten Zustand wieder
20 gestartet wird.
2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß der Zustand des wenigstens einen Zeitmessers (15) bei einer Unterbrechung der ersten Task möglichst früh gespeichert,
25 und nach Beendigung der Unterbrechung möglichst spät restauriert wird.
3. Verfahren nach einem der Ansprüche 1 oder 2, dadurch gekennzeichnet, daß im Falle einer Unterbrechung der ersten
30 Task aufgrund einer prioritätshöheren zweiten Task die Laufzeit der zweiten Task während der Unterbrechung der Laufzeit der ersten Task gemessen wird.

- 12 -

4. Verfahren nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, daß sämtliche Laufzeitmessungen sämtlicher Tasks unter Verwendung von genau einem Zeitmesser durchgeführt werden, dessen Speicherkapazität der Anzahl
5 der unterschiedlichen Prioritätsebenen der Tasks angepaßt ist.

5. Vorrichtung zur Messung der Laufzeit wenigstens einer Task in einem eine Anzahl von Tasks umfassenden
10 Echtzeitsystem, mit wenigstens einem Zeitmesser (15) und Mitteln (18) zum Starten und Stoppen des Zeitmessers (15), dadurch gekennzeichnet, daß dem Zeitmesser (15) ein Speicher (16) mit einer Speicherkapazität, die der Anzahl der Prioritätsebenen der Tasks angepaßt ist, zugeordnet
15 ist.

6. Vorrichtung nach Anspruch 5, dadurch gekennzeichnet, daß der Zeitmesser ein Hardwaretimer ist.

20 7. Vorrichtung nach einem der Ansprüche 5 oder 6, dadurch gekennzeichnet, daß die Schrittweite des Zeitmessers (15) wesentlich kleiner als die übliche Laufzeit der Tasks ist.

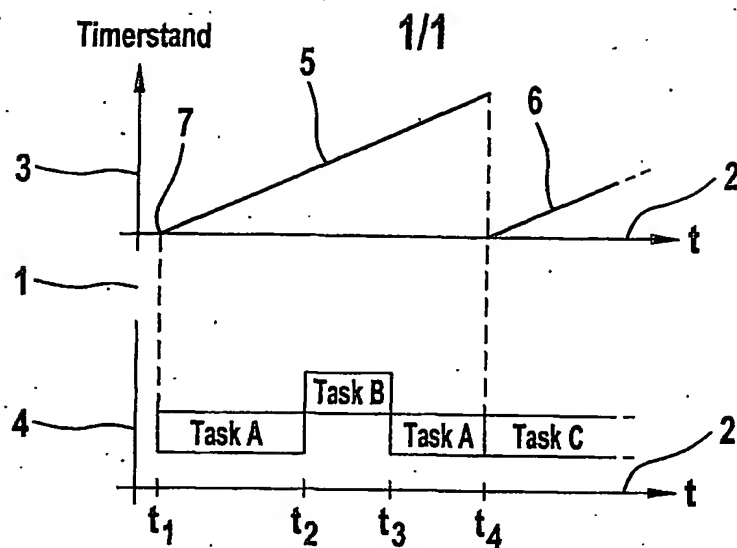


FIG. 1

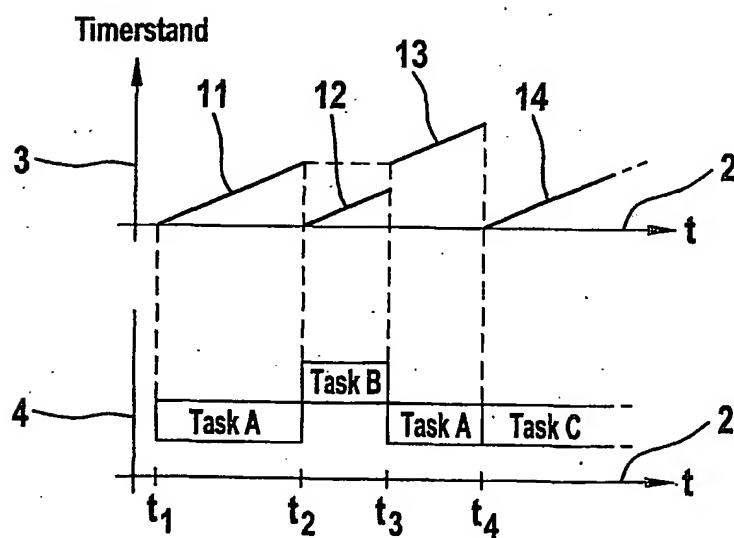


FIG. 2

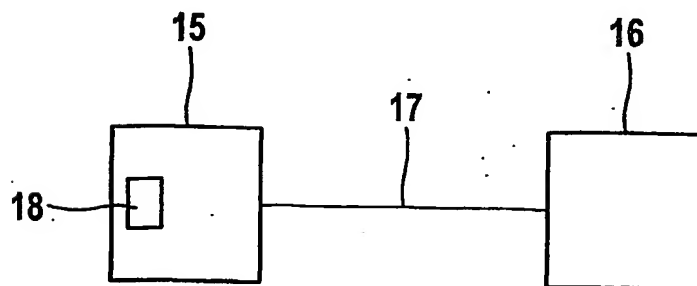


FIG. 3